AD-A067 188 GEORGE WASHINGTON UNIV WASHINGTON D C INST FOR MANAG--ETC F/G 12/1
COMPUTING EQUILIBRIA VIA NONCONVEX PROGRAMMING.(U)
NOV 78 J F BARD, J E FALK DAAG29-76-G-0150
UNCLASSIFIED SERIAL-T-386 ARO-13622.6-M NL

1 OF 1
AD
A067188

END
DATE
FILMED
6-79
DDC

ARO 13622.6-M

LEVEL II

AD A067188

DDC FILE COPY

THE
GEORGE
WASHINGTON
UNIVERSITY

12

STUDENTS FACULTY STUDY R
ESEARCH DEVELOPMENT FUT
URE CAREER CREATIVITY CO
MMUNITY LEADERSHIP TECH
NOLOGY FRONTI ESIGN
ENGINEERING APP ENC
GEORGE WASHIN NIV

79 04 09 152

INSTITUTE FOR MANAGEMENT
SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

LEVEL II

(12)

AD A067188

DDC FILE COPY

COMPUTING EQUILIBRIA VIA NONCONVEX PROGRAMMING

by

Jonathan F. Bard
James E. Falk

Serial T-386
10 November 1978

The George Washington University
School of Engineering and Applied Science
Institute for Management Science and Engineering

Research Sponsored by
U.S. Army Research Office
Contract DAAG29-76-G-0150

79 04 09

None 18 ARO 19 13622.6-M

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. ~~REPORT NUMBER~~ 14 SERIAL-T-386 / 2. ~~GOVT ACCESSION NO.~~ 9 Scientific rept.	3. ~~RECIPIENT'S CATALOG NUMBER~~
4. TITLE *(and Subtitle)* 6 COMPUTING EQUILIBRIA VIA NONCONVEX PROGRAMMING .	5. ~~TYPE OF REPORT & PERIOD~~ COVERED SCIENTIFIC
	6. PERFORMING ORG. REPORT NUMBER T-386
7. AUTHOR(s) 10 JONATHAN F. BARD, JAMES E. FALK	8. CONTRACT OR GRANT NUMBER(s) 15 DAAG29-76-G-0150
9. PERFORMING ORGANIZATION NAME AND ADDRESS THE GEORGE WASHINGTON UNIVERSITY INST. FOR MANAGEMENT SCIENCE & ENGINEERING WASHINGTON, D. C. 20052	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. ARMY RESEARCH OFFICE BOX 12211 RESEARCH TRIANGLE PARK, N. C. 27709	12. ~~REPORT DATE~~ 11 10 Nove[redacted]78
	13. NUMBER OF PAGES 39
14. MONITORING AGENCY NAME & ADDRESS *(if different from ~~Controlling Office~~)* 12 42p	15. SECURITY CLASS. *(of this report)* NONE
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

DISTRIBUTION OF THIS REPORT IS UNLIMITED.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

ECONOMIC EQUILIBRIUM
TRAFFIC FLOW EQUILIBRIUM
NONCONVEX PROGRAMMING
COMPLEMENTARITY THEORY

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The problem of determining a vector that places a system in a state of equilibrium is studied with the aid of mathematical programming. The approach derives from the logical equivalence between the general equilibrium problem and the complementarity problem, the latter being explicitly concerned with finding a point in the set next page

$$S = \{x : \langle x, g(x)\rangle = 0 \ , \ g(x) \leqq 0 \ , \ x \geqq 0\}$$

(over)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

406 743 Gen

20. Abstract (continued)

An associated nonconvex program, $\min\{-\langle x,g(x)\rangle : g(x) \leq 0, x \geq 0\}$, is proposed whose solution set coincides with S. When the excess demand function $g(x)$ meets certain separability conditions, equilibrium solutions are obtained by using an established branch and bound algorithm. Because the best upper bound is known at the outset, an independent check for convergence can be made at each iteration of the algorithm, thereby greatly increasing its efficiency. A number of examples drawn from economic and network theory are presented in order to demonstrate the computational aspects of the approach. The results appear promising for a wide range of problem sizes and types, with solutions occurring in a relatively small number of iterations.

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Abstract
of
Serial T-386
10 November 1978

COMPUTING EQUILIBRIA VIA NONCONVEX PROGRAMMING

by

Jonathan F. Bard*
James E. Falk

The problem of determining a vector that places a system in a state of equilibrium is studied with the aid of mathematical programming. The approach derives from the logical equivalence between the general equilibrium problem and the complementarity problem, the latter being explicitly concerned with finding a point in the set

$$S = \{x : \langle x, g(x) \rangle = 0 \ , \ g(x) \leqq 0 \ , \ x \geqq 0\}$$

An associated nonconvex program, $\min\{-\langle x, g(x)\rangle : g(x) \leqq 0, x \geqq 0\}$, is proposed whose solution set coincides with S . When the excess demand function $g(x)$ meets certain separability conditions, equilibrium solutions are obtained by using an established branch and bound algorithm. Because the best upper bound is known at the outset, an independent check for convergence can be made at each iteration of the algorithm, thereby greatly increasing its efficiency. A number of examples drawn from economic and network theory are presented in order to demonstrate the computational aspects of the approach. The results appear promising for a wide range of problem sizes and types, with solutions occurring in a relatively small number of iterations.

*The Aerospace Corporation

Research Sponsored by
U. S. Army Research Office
Research Triangle Park, North Carolina 27709

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Program in Logistics

COMPUTING EQUILIBRIA VIA NONCONVEX PROGRAMMING

by

Jonathan F. Bard
James E. Falk

1.0 Introduction

In this paper we investigate a procedure for computing equilibria from the vantage point of mathematical programming. A competitive model of an economy will serve as the basis for the discussion although a variety of contexts would have been equally suitable. Other types of equilibrium problems, such as those arising in traffic network analysis, have direct conceptual and analytic counterparts to those found in economics, and are hence amenable to the same solution techniques.

A state of equilibrium exists when competing or opposing forces are brought into balance. One of the major themes of economic theory is that the behavior of a complex economic system can be viewed as an equilibrium arising from the interaction of a number of economic units, each motivated by their own special interests. General equilibrium theory ([2], [22], [24]) seeks to determine the point at which this balance can be struck, and in so doing focuses on the interrelationships that exist among the markets for goods and services in the economy. The analysis, however, is carried out in terms of individual decision makers and commodities rather than in terms of aggregates. The fundamental questions that general equilibrium theory attempts to answer are the same as those posed in macroeconomic theory: given different economic environments, what goods will the economy produce, how will these be produced, and who will obtain them? But, where macroeconomics provides answers in terms of aggregates, general equilibrium theory provides answers in terms of the individual consumers, producers, and commodities making up these aggregates.

Consider, for the moment, a model in which m consumers are engaged in the exchange of commodities which they initially own and in which production or supply is ignored. Suppose there are n goods in the economy and that each of the consumer's preferences is represented by a utility function. A bundle of goods x is preferred to a bundle x' by consumer i $(i = 1,\ldots,m)$ if and only if $u_i(x) > u_i(x')$ where the utility function $u_i : R^n \to R$ is generally assumed to be strictly concave and continuous. Let $p \in R^n$ be the vector of prices for the n goods. The demands of the ith consumer are determined by the solution to the following problem:

$$\begin{aligned} &\text{maximize} \quad u_i(x) \\ &\text{subject to} \quad \langle p,x\rangle \leqq \langle p,w^i\rangle \\ &\qquad\qquad\qquad x \geqq 0 \end{aligned}$$

where w^i is the initial wealth or resource endowment of the ith consumer, $i = 1,\ldots,m$. We shall assume that the solution vector for this problem, $d^i(p)$, can be written as a continuous function of the prices p . The individual trader's excess demand function is $d^i(p) - w^i$ $(i = 1,\ldots,m)$ and will be denoted by $g^i(p)$. The excess demand will be positive for those commodities whose stock he wishes to increase by exchange and negative for the remaining items. If it is assumed that all purchases are to be financed solely by the sale of assets, then individual budgetary constraints lead to the following identity:

$$p_1 d_1^i(p) + \cdots + p_n d_n^i(p) = p_1 w_1^i + \cdots + p_n w_n^i \ . \qquad (1)$$

The market excess demand function $g : R^n \to R^n$ is simply the sum of the individual excess demand functions

$$g(p) = \sum_{i=1}^{m} (d^i(p) - w^i) \ .$$

An equilibrium price vector p^* is one for which all of the market excess demands are less than or equal to zero with a zero price for any commodity whose excess demand is strictly less than zero. This leads to the formulation of the complementarity problem ([8], [17]):

$$g(p) \leqq 0 \ , \tag{2}$$

$$p \geqq 0 \ , \tag{3}$$

$$\langle p, g(p) \rangle = 0 \ , \tag{4}$$

whose solution p^* will be the focal point of this paper. Condition (4), known as the Walras Law [27], is obtained by aggregating individual excess demands to create the set of market excess demand functions, and then put in the form of (1). Walras' law holds for all price vectors p whether they be in equilibrium or not.

We note here that production may easily be incorporated in this model by either replacing or augmenting the ith consumer's initial wealth w^i by a supply function. For individual i, this function relates the prevailing market prices p to the quantity of goods produced.

A number of persons, including Nash [20], Arrow and Debreu [1], and Kuhn [16] have studied the existence problem of the competitive model from the standpoint of combinatorial topology. The first algorithms, however, actually designed for computing economic equilibria were developed by Scarf [23], and were based on a procedure for approximating a fixed point of a continuous mapping. More recently, Wilson [28] and Elken [10] have exploited path methods in the pursuit of greater computational efficiency. In a slightly different vein, Lemke [17] offered some constructive proofs relating to the existence of equilibrium points for bimatrix games. His work strongly suggested a computational scheme for models with linear excess demand functions.

This paper presents an alternative procedure for computing equilibria for a class of problems where the excess demand function or its logical equivalent has an explicit representation that can be converted to a separable form. Solutions are obtained by first recasting the complementarity problem into a nonconvex minimization problem whose optimal value or best upper bound is known at the outset, and then using Falk's [12] algorithm to locate a global solution. This allows us to go beyond the common linear formulations of an economy or network (e.g., see Eaves [8], Negishi [21], or Asmuth, Eaves, and Peterson [4]) which, *in spite of their outward* simplicity, must appeal to rather complicated algorithms if solutions are to be obtained.

The algorithm which we subsequently describe and use as an alternative for solving (2) - (4) is based on a branch and bound philosophy, and as such, computes a convergent sequence of upper and lower bounds on the optimal value of the problem. In our case, however, because the best upper bound on the objective function is known to be zero, the amount of work necessary to achieve convergence is significantly reduced. The usual requirement of finding a point that yields equality between the best upper and best lower bounds is replaced by the simpler requirement of finding any point that yields an objective value of zero.

In the next section, the complementarity problem is reformulated as a nonconvex minimization problem whose solution yields the desired equilibrium vector. Next, the method is applied to a number of sample problems and our computational experience is detailed. Here we see that the results are obtained in a surprisingly small number of iterations of the algorithm.

2.0 Reformulation of the Complementarity Problem

In the complementarity problem derived above, there is no objective function to be optimized. Indeed, in many complex economic equilibrium problems there does not appear to be a "natural" objective function whose optimization yields prices and quantities in equilibrium (see, e.g., Scarf [24]).

In spite of this, consider the following "artificial" minimization problem P:

$$v^* = \min\{-\langle p, g(p)\rangle : g(p) \leqq 0 \;,\; p \geqq 0\} \tag{P}$$

Now let p^* be a solution of the complementarity problem (i.e., p^* is a vector of equilibrium prices). Then p^* is feasible to problem P, and yields a value of 0 to the objective function. Since this objective function is greater than or equal to zero at all feasible points, $v^* = 0$. Conversely, it is clear that any solution of (P) for which $v^* = 0$ must be a vector of equilibrium prices.

Problem P is of a nonconvex nature, and in general, no suitable technique exists for the determination of a global, rather than a local solution; however, if each excess demand function g_i, $i = 1,2,\ldots n$, is separable, i.e.,

$$g_i(p) = \sum_{j=1}^{n} g_{ij}(p_j) \quad i = 1,2,\ldots,n$$

and each g_{ij} is continuous, then (P) can be written as a separable programming problem whose approximate global solution can be obtained with arbitrary precision.

We now formulate an equivalent problem with a different objective function but the same constraint region whose optimal value is equal to that of (P). The equivalent problem P' is

$$\min\{\sum_i \min(p_i,-g_i(p)) : p \geqq 0 \text{ , } g(p) \leqq 0\} \tag{P'}$$

Rewriting the objective function in (P'), we get the desired result:

$$\min_{\substack{g(p)\leqq 0\\ p\geqq 0}} \{\sum_i (\min(0,-g_i(p)-p_i) + p_i)\}$$

$$= \min_{\substack{g(p)\leqq 0\\ w+p+g(p) = 0\\ p\geqq 0}}\{\sum_i (\min(0,w_i) + p_i)\}$$

$$= \min_{\substack{\sum_j g_{ij}(p_j) \leqq 0 \\ w_i+p_i+\sum_j g_{ij}(p_j) = 0 \quad i = 1,2,\ldots,n\\ p\geqq 0}}\{\sum_i (\min(0,w_i) + p_i)\} \tag{S}$$

where the w_i's will be referred to as auxiliary variables.

Problem S is still a nonconvex programming problem, but its separable structure, created at the expense of a twofold increase in dimensionality, makes its mathematics much more tractable. The traditional method for treating separable problems involves calculating piecewise linear approximations of the associated functions and applying a modification of the simplex method to the resulting problem (see, e.g., Miller [19]). The modification amounts to a restriction on the usual manner of selecting variables to exchange roles (basic to nonbasic and vice versa) and will yield a local but not necessarily a global solution of the approximating problem.

An algorithm for finding global solutions of nonconvex separable problems was developed by Falk and Soland [13] and Soland [25]. The method is based on the branch and bound philosophy and yields a (generally infinite) sequence of points whose cluster points are global solutions of the problem. The implementation of the method is limited by the necessity of computing convex envelopes [11] of the functions involved although a number of applications have been shown possible when these functions exhibit special structures (e.g., concave or piecewise linear).

The inherent limitations that special problem structures impose have been overcome by the introduction of two algorithms independently developed by Beale and Tomlin [5] and Falk [12]. For this paper, we have used the programming code MOGG based on the algorithm proposed by Falk and written by Grotte [15], to solve a number of equilibrium problems. The results are presented in the next section.

3.0 Computational Experience

A variety of equilibrium problems have been studied to test the approach outlined above. The first is a multicommodity, transshipment problem defined on an affine network, taken from Asmuth, Eaves, and Peterson [4] who used Lemke's algorithm [17] to obtain a solution. The second involves a simple competitive market comprising three producers, three consumers, and three commodities. The supply and demand functions in this economy are given a piecewise linear formulation, and three equilibrium points are known to exist. The third problem is identical to the second except a majority of the piecewise linear functions have been recast as continuous, smooth functions. The fourth problem provides an example outside the context of economics, and is derived from a 3 node, 4 arc traffic network whose equivalent excess demand function is both nonlinear and nonseparable.

The algorithm itself is based on branch and bound techniques and considers subsets of a linear polyhedron containing the feasible region of (S). A lower bound on the optimal value of the problem is found by minimizing the objective function over each of these subsets and selecting the smallest value obtained. A check for the solution is made which, if successful, yields a global solution of the piecewise linear approximation to (S). If the check fails, the subset corresponding to the smallest lower bound is further subdivided into either two or three new linear polyhedra and the process continues as before with new and sharper bounds being determined. The process is finite and terminates with a global solution of the approximate problem.

3.1 Transshipment on Affine Networks - Economic equilibria on certain affine, multicommodity, transshipment networks were first studied by the regional economists Taykayama and Judge [26] using quadratic programming. Recently, Asmuth, Eaves, and Peterson [4] have constructed a more general approach that utilizes the economic equilibrium conditions directly without first passing to a quadratic programming problem. A brief discussion of their model and the solution to the sample problem presented in their paper follow.

The transshipment problem can conveniently be represented by a directed graph (N,L) with a finite number of nodes (members of N) and links (members of L) on which a finite number of commodities can be transported. Each node i in N represents the set of producers and/or consumers at a specific spatial location; and each link ij in L represents a specific facility for transporting commodities from some node i to a different node j . (In particular, we assume that there are no loops; i.e., no links connecting a given node to itself.) Each link is aligned to coincide with a direction of possible transport; therefore, at least two links must connect nodes between which commodities can be transported in either direction.

The nodes are enumerated in any order, consecutively beginning with one - as illustrated by the graph in Figure 1.

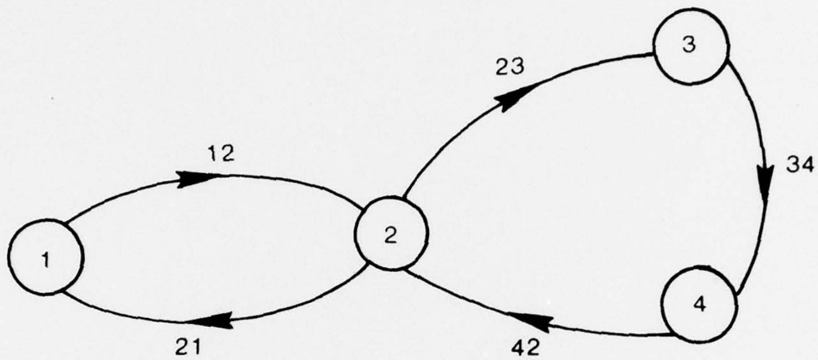

Figure 1. Sample Transshipment Network.

Let p_d^i and p_s^i be n vectors denoting the unit demand price and the unit supply price of n commodities at node i, and let $p^{ij} \in R^n$ denote the cost of transporting each of these n commodities over link ij. Affine relations are assumed between prices and quantities; i.e.,

$$p^i = A^i x^i + a^i \quad \text{for each node } i \in N \tag{5}$$

where $p^i = (p_d^i, p_s^i)$, A^i and a^i are given constant matrices and vectors (which arose in [4] from inverting the difference between given supply and demand quantities, originally expressed as affine functions of price), and x^i is an n-dimensional vector representing the excess quantity of each commodity produced by node i. It is also assumed that

$$p^{ij} = A^{ij}x^{ij} + a^{ij} \quad \text{for each link } ij \in L \tag{6}$$

where A^{ij} and a^{ij} are given constant matrices and vectors (which arose from describing the transport prices as functions of transport volumes), and $x^{ij} \in R^n$ denotes the quantity of n commodities transported over link ij .

The quantities are constrained by the nonnegativity condition

$$x^{ij} \geqq 0 \quad \text{for each link } ij \in L \tag{7}$$

and the commodity conservation condition

$$x^i = \sum_{j \in N} x^{ij} - \sum_{j \in N} x^{ji} \quad \text{for each node } i \in N \tag{8}$$

where $x^{ij} = 0$ if $ij \notin L$. Note that although x^{ij} is nonnegative by virtue of the choice of link direction, the components of x^i might be positive or negative depending on whether node i is a net exporter or net importer of a particular commodity.

The price stability condition on p leads to the following relationship:

$$p^{ij} + p_s^i \geqq p_d^j \quad \text{for each link } ij \in L \tag{9}$$

To see this, assume that (9) is violated for some commodity c . As a consequence some economic agent would find it profitable to purchase as much of commodity c as possible at node i and transport it over link ij for resale at node j . This would clearly be an economically unstable situation.

The final relationship needed to establish equilibrium is the complementarity condition

$$< x^{ij}, (p_d^j - p^{ij} - p_s^i) > = 0 \quad \text{for each link } ij \in L . \tag{10}$$

This condition is imposed to insure that no positive flow x^{ij} will occur on a link if the cost $p^{ij} + p_s^i$ of a commodity at node j exceeds the price p_d^j which a consumer is willing to pay.

To conform with the notation developed in Section 1.0, a function $g^{ij} \varepsilon R^n$ equivalent to the excess demand function but now expressed in terms of prices rather than quantities will be defined by the following expression:

$$g^{ij} \doteq p_d^j - p^{ij} - p_s^i \quad \text{for each link } ij \varepsilon L$$

Through the appropriate substitutions the solution (x,p) to the economic equilibrium conditions (5) - (10) can be described entirely in terms of the solution (x,g) to the linear complementarity conditions

$$x \geqq 0 \quad \langle x,g \rangle = 0 \quad g \leqq 0 \tag{11}$$

$$g = - Mx - v \tag{12}$$

where x , g , and v are vectors equal in size to the number of links times the number of commodities, and M is a square matrix of comparable dimension whose components are given in Figure 2. The constant v follows from the substitution of (5) and (6) into (10) and is given by

$$v^{ij} = a_d^j - a^{ij} - a_s^i \quad \text{for each link } ij \varepsilon L$$

For purposes of illustration, the 2-commodity, 5-link network shown in Figure 1 has been considered for analysis. The reworked data for this problem is displayed in Figure 3. If conditions (11) and (12) are now put into the format of (S) we get a problem of the form

$$\min_{x,w} \sum_{i=1}^{10} (\min(0,w_i) + x_i) \tag{A}$$

$$w_i - M_i x + x_i = v_i$$

$$- M_i x \leqq v_i \qquad i = 1,2,\ldots,10$$

$$x \geqq 0$$

where M_i is the ith row of M .

The algorithm that is used for the computations does not solve the original problem S, but constructs an approximate problem to solve by replacing each of the associated functions with their piecewise linear

$$
[M \,\|\, v] = \left[\begin{array}{c|c|c|c|c||c}
A_1 + A_2 + A^1 & -(A_1 + A_2) & -A_2 & 0 & A_2 & V^{12} \\ \hline
-(A_1 + A_2) & A_1 + A_2 + A^2 & A_2 & 0 & -A_2 & V^{21} \\ \hline
-A_2 & A_2 & A_2 + A_3 + A^3 & -A_3 & -A_2 & V^{23} \\ \hline
0 & 0 & -A_3 & A_3 + A_4 + A^4 & -A_4 & V^{34} \\ \hline
A_2 & -A_2 & -A_2 & -A_4 & A_2 + A_4 + A^5 & V^{42}
\end{array}\right]
$$

Figure 2. Constituents of Matrix for Sample Network

$$
[M \,\|\, v] = \left[\begin{array}{cc|cc|cc|cc|cc||c}
4 & -1 & -2 & 1 & -1 & 1 & 0 & 0 & 1 & -1 & -1 \\
2 & 3 & -2 & -2 & 0 & -1 & 0 & 0 & 0 & 1 & 2 \\ \hline
-2 & 1 & 3 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & -1 \\
-2 & -2 & 2 & 3 & 0 & 1 & 0 & 0 & 0 & -1 & -5 \\ \hline
-1 & 1 & 1 & -1 & 3 & -3 & -1 & 1 & -1 & 1 & -2 \\
0 & -1 & 0 & 1 & 2 & 3 & -1 & -1 & 0 & -1 & 1 \\ \hline
0 & 0 & 0 & 0 & -1 & 1 & 2 & -2 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 4 & 2 & -2 & 0 & -3 \\ \hline
1 & -1 & -1 & 1 & -1 & 1 & 0 & 2 & 2 & -3 & 2 \\
0 & 1 & 0 & -1 & 0 & -1 & -2 & 0 & 4 & 2 & 4
\end{array}\right]
$$

Figure 3. Data for Affine Network

convex envelopes. A related problem is simultaneously introduced which gives a sharper underestimate of the optimal value of the approximating problem than does the convex envelope problem. It is this related problem that the branch and bound procedure solves first to get estimates on the optimal value of the approximating problem, and to set up new problems if the estimates do not yield a global solution.

The functions defining the constraint region of problem A are all linear, and hence convex, and therefore, will not be replaced in the approximate problem. The functions associated with the nonlinear variables w_i in the objective function (i.e., $\min(0,w_i)$, $i = 1,\ldots,n$) are piecewise linear, but concave and will be replaced in the approximate problem by their convex envelopes, which in this case are straight lines. This is illustrated in Figure 4.

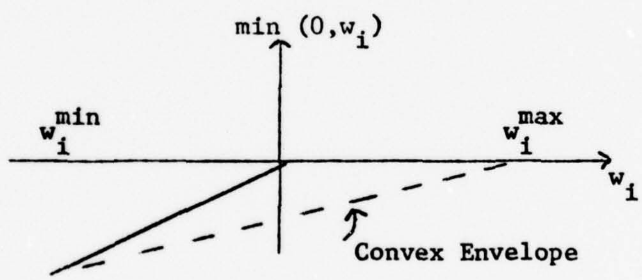

Figure 4. Convex Envelope of $\min(0,w_i)$.

The branch and bound technique proceeds to divide the domain of these functions into pieces corresponding to their linear segments and separately solves the set of related problems in which the nonlinear variables are respectively limited. When every function is piecewise linear, as is the case with (A), we get an exact solution to the original problem.

It is customary with branch and bound methods to describe the algorithm in terms of a branch and bound tree. The nodes of the tree correspond to the related linear subproblems, while the branches of the tree correspond to the set on which the branching variables are defined. A solution is obtained when the best upper bound at any node is equal to zero. That is, any feasible point yielding a subproblem value of zero necessarily satisfies the equilibrium conditions. For problem A, no tree developed because the solution was obtained on the first iteration of the algorithm. The optimal vector x is given by:

$$x^1 = (0.2353,\ 0.7059) \quad x^2 = (0.0,\ 2.2941) \quad x^3 = (1.5294,\ 0.0)$$

$$x^4 = (1.0098,\ 0.2151) \qquad x^5 = (0.0,\ 0.0)\ .$$

The formulation of (A) required the addition of 10 auxiliary variables to the original set of 10 linear variables. The former were each divided into two intervals for the purposes of branching. This division, corresponding to the segments of the piecewise linear functions defined for these variables, implied that any branch and bound tree produced by the algorithm could be at most 10 branches deep and that no variable could appear more than once along any path. In theory, it might have been necessary to solve up to $2^{11} - 1$ subproblems before reaching a solution; however, the fact that the first subproblem produced an equilibrium point underscores the computational efficiencies that result from having available at the outset a means of independently checking each iteration for convergence.

Each subproblem solved by MOGG is a linear program. When the excess demand functions are affine, the solution vector necessarily yields a feasible point to the original problem. The upper bound associated with this feasible point will always be greater than or equal to zero, but generally not correspond to an equilibrium solution. Mangasarian [18] has shown that the linear complementarity problem is equivalent to a linear program whose cost coefficients are dependent upon the structure of M. The similarity between these linear programs and those set up by MOGG when M meets certain conditions admits the possibility that MOGG will produce an equilibrium point on the first iteration. Although these conditions might logically arise in some economic contexts, they were not present in this example and, hence, did not influence the rate of convergence.

3.2 <u>A Piecewise Linear Market</u> - This example [7] provides a simple explanation of how a competitive market operates. As is common in microeconomic theory, we will distinguish among individuals according to the economic functions that they perform or on the basis of the kinds of decisions they make. Thus, a consumer is an individual (or unit) that consumes commodities and supplies inputs to production. The role of the consumer may be defined as that of choosing from among the alternative commodity bundles available to him. Similarly, a producer is an individual (or group) that utilizes inputs to produce commodities. The role of the producer may be characterized as that of choosing from among the alternative input-output patterns available to him. The same individual might appear in the economy both as a consumer and as a producer. Once the choices are made, a state of the economy is defined.

Under certain assumptions (see, Quirk and Saposnik [22]) for a market that contains n commodities, m consumers, and ℓ producers, the aggregated (net) amounts of commodities demanded and supplied for any vector of prices can be determined by a simple summation of the amounts demanded and supplied by individual consumers and producers. Thus, given the price vector p, where $p = (p_1, p_2, \ldots, p_n)$, we can write $x_{ij}(p)$ as the amount of the ith commodity consumed (or supplied as an input in production) by the jth individual at the set of prices given by p; and $y_{ik}(p)$ as the amount of the ith commodity produced (or used up as an input in production) by the kth firm at the set of prices given by p. Then, the aggregate (net) consumption of commodity i by consumers is given by

$$x_i(p) = \sum_{j=1}^{m} x_{ij}(p) \qquad i = 1,2,\ldots,n$$

and the aggregate (net) production of commodity i by producers is given by

$$y_i(p) = \sum_{k=1}^{\ell} y_{ik}(p) \qquad i = 1,2,\ldots n$$

We then define $x(p)$ and $y(p)$ as point to point mappings from R^n into itself. In the absence of any initial endowment the excess demand function can be written as $g(p) = x(p) - y(p)$.

The sample economy under consideration contains three commodities, three consumers, and three producers. The associated supply and demand functions are assumed to be piecewise linear, and are given in graphic form in Figures 5 and 6. To conform with the presentation in [7], the equilibrium quantity rather than the equilibrium price will be computed. The following notation will be used:

$p_{s_{ij}}$ = jth producer's supply price for commodity i

$p_{d_{ij}}$ = jth consumer's demand price for commodity i

x_i = quantity of ith commodity consumed

y_i = quantity of ith commodity produced

where $i,j = 1,2,3$, $p_{s_i} = \sum_j p_{s_{ij}}$ is a function of the consumption variable x and $p_{d_i} = \sum_j p_{d_{ij}}$ is a function of the production variable y .

An equilibrium point will exist if the following conditions are satisfied:

$$x \geqq 0 \text{ , } y \geqq 0 \tag{13}$$

$$x - y = 0 \tag{14}$$

$$p_d - p_s \leqq 0 \tag{15}$$

$$\langle x, p_d - p_s \rangle = 0 \tag{16}$$

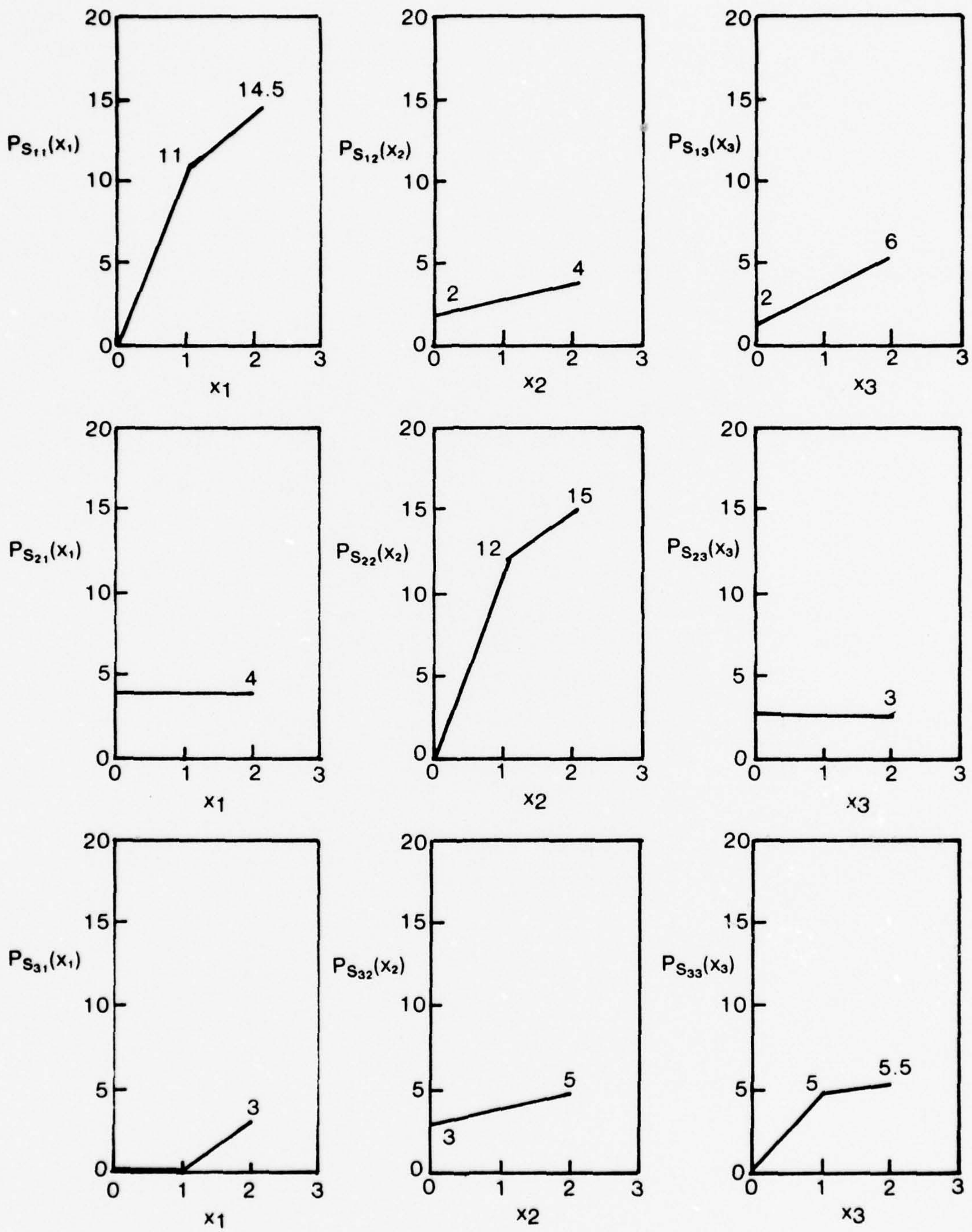

Figure 5. Supply Functions for Piecewise Linear Market

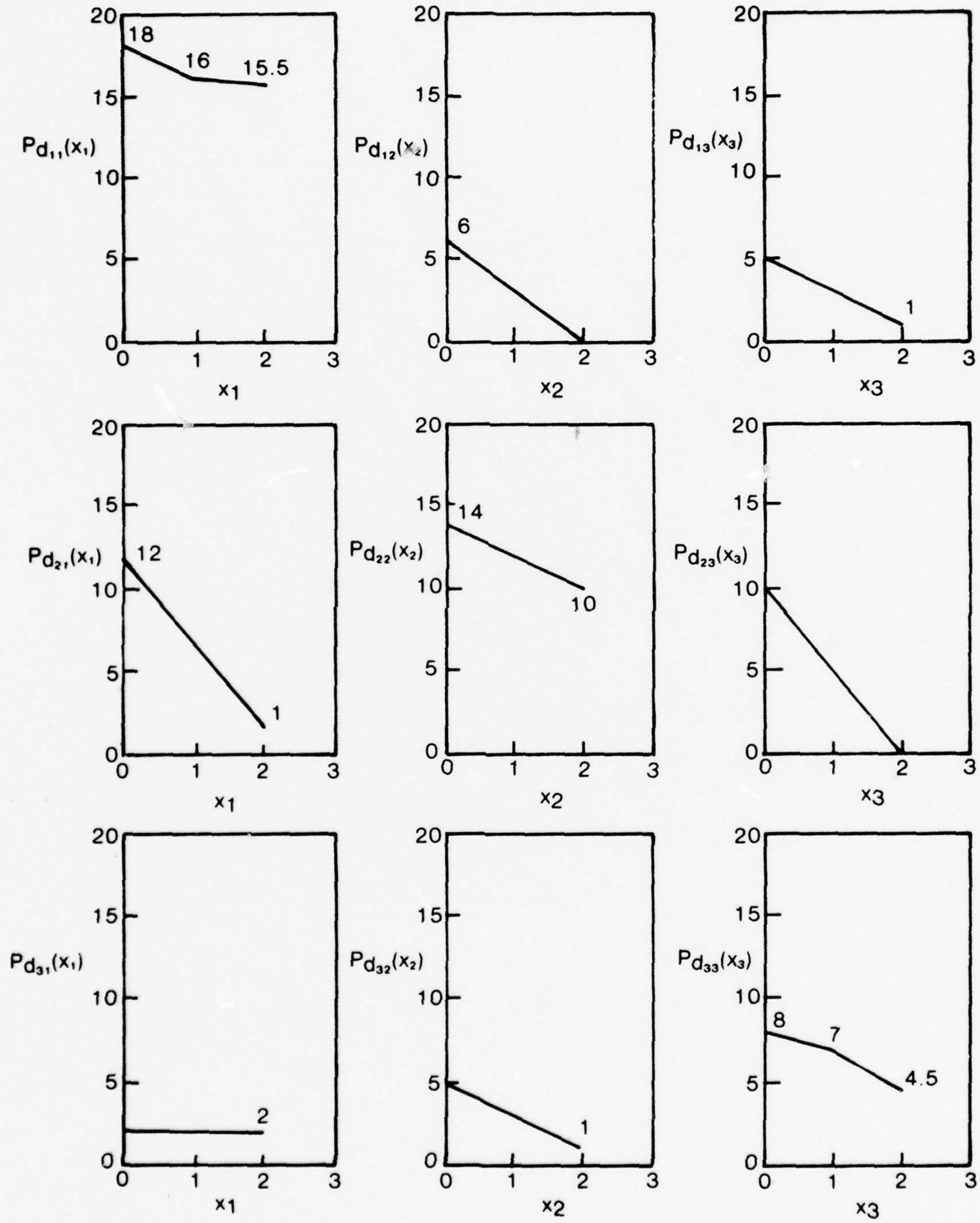

Figure 6. Demand Functions for Piecewise Linear Market

where p_s and p_d are the three-dimensional market supply price and market demand price vectors. The first condition assures feasibility; the second condition clears the market; the third condition assures price consistency by requiring the excess demand function to be less than or equal to zero; and the fourth condition is Walras' Law and reflects the following circumstances: if x_i, the quantity of the ith commodity being purchased, was positive and if the producers' supply price p_{s_i} was greater than the consumers' demand price p_{d_i}, then the producers would be losing money and begin to lower y_i $(=x_i)$ to zero. Such a situation would be economically unstable.

Conditions (13) - (16) can now be written as a minimization problem in the form of (P).

$$\min\{\langle -x\,,\,(p_d - p_s)\rangle\}$$
$$p_d - p_s \leqq 0$$
$$x \geqq 0$$

In order to recast this problem in the form of (S), the consumption and production data given in Figures 5 and 6 must be aggregated over their respective agents to obtain the market demand and supply curves p_d and p_s. This has been done for each of the three commodities.

Commodity 1

$$p_{d_1} = \begin{cases} -2x_1+18 & ;\ x_1 \leqq 1 \\ -0.5x_1+16.5; & x_1 > 1 \end{cases} + (6-3x_2) + (5-2x_3)$$

$$p_{s_1} = \begin{cases} 11x_1 & ;\ x_1 \leqq 1 \\ 7.5x_1 - 3.5; & x_1 > 1 \end{cases} + (2+x_2) + (2+2x_3)$$

Commodity 2

$$p_{d_2} = (12-5x_1) + (14-2x_2) + (10-5x_3)$$

$$p_{s_2} = 4 + \begin{Bmatrix} 12x_2 \ ; & x_2 \leqq 1 \\ 3x_2+9; & x_2 > 1 \end{Bmatrix} + 3$$

Commodity 3

$$p_{d_3} = 2 + (5-2x_2) + \begin{Bmatrix} -x_3 + 8 & ; x_3 \leqq 1 \\ -2.5x_3 + 9.5; & x_3 > 1 \end{Bmatrix}$$

$$p_{s_3} = \begin{Bmatrix} 0 & ; x_1 \leqq 1 \\ 3x_1 - 3; & x_1 > 1 \end{Bmatrix} + (3+x_2) + \begin{Bmatrix} 5x_3 & ; x_3 \leqq 1 \\ 0.5x_3 + 4.5; & x_3 > 1 \end{Bmatrix}$$

The minimization problem in its separable form becomes

$$\min_{x,w} \sum_{i=1}^{3} \{\min(0,w_i) + x_i\} \qquad \text{(B)}$$

$$\left.\begin{array}{l} w_i + p_{d_i} - p_{s_i} + x_i = 0 \\ p_{d_i} - p_{s_i} \leqq 0 \end{array}\right\} \quad i = 1,2,3$$

$$x \geqq 0$$

where p_{d_i} and p_{s_i} are defined above.

Each of the six variables in (B) is nonlinear, the first three (w_1,w_2,w_3) corresponding to the auxiliary variables and the second three (x_1,x_2,x_3) to the original problem variables. The associated functions are piecewise linear and contain at most one break point. This means that the branch and bound tree can be at most six nodes deep and that a maximum of $2^7 - 1$ subproblems might have to be solved. Once again though, the algorithm converged on the first iteration. The computed best upper bound for the first subproblem is zero and hence the solution.

If the algorithm is permitted to run past this point until its usual termination conditions are met a total of 19 subproblems will be solved. Figure 7 depicts the resultant branch and bound tree which serves to illustrate both the advantage of knowing the optimal value at the outset and the amount of work required to search for alternative solutions. The known results are corroborated at node 9.2 where roundoff errors have produced a best upper bound of -0.2×10^{-4} and a best lower bound of 0.1×10^{-5}, a minor contradiction. The two numbers adjacent to each node represent the best upper and lower bounds for that subproblem. A bar in the place of the best upper bound indicates that no corresponding feasible point to the approximate problem exists. The numbers along the branches refer to the branching variables associated with the preceding node, and the + and - signs indicate whether the particular auxiliary variable was permitted to range over the set of positive real numbers or negative real numbers, respectively. The bars appearing below the nodes indicate that either the lower bounds of the associated subproblems are all greater than the current best upper bound or that they are infeasible and, therefore, cannot contain the solution.

In terms of the actual variables, the solution vectors are $x^* = (2,1,1)$ and $w^* = (-2,-1,-1)$. From the equality constraints in (B), it can be seen that $x^* + w^* = 0$ whenever the corresponding excess demand functions are binding. By tracing the convergent path backwards from node 9.2 to node 0.0 we see that the branches that fall along this path correspond to the nonpositive orthant of w. The subscripts attached to the branching variables in the tree denote the (closed) intervals over which the original problem variables are defined for all subsequent subproblems.

3.3 A General Market - The separable programming algorithm works by first replacing each of the original problem functions with their piecewise linear convex envelopes, and then creating a new problem to solve as an ultimate approximation. From this approximate problem a series of convex subproblems issue that are set up and solved under the branch and bound philosophy. If the original functions are all piecewise linear (but not

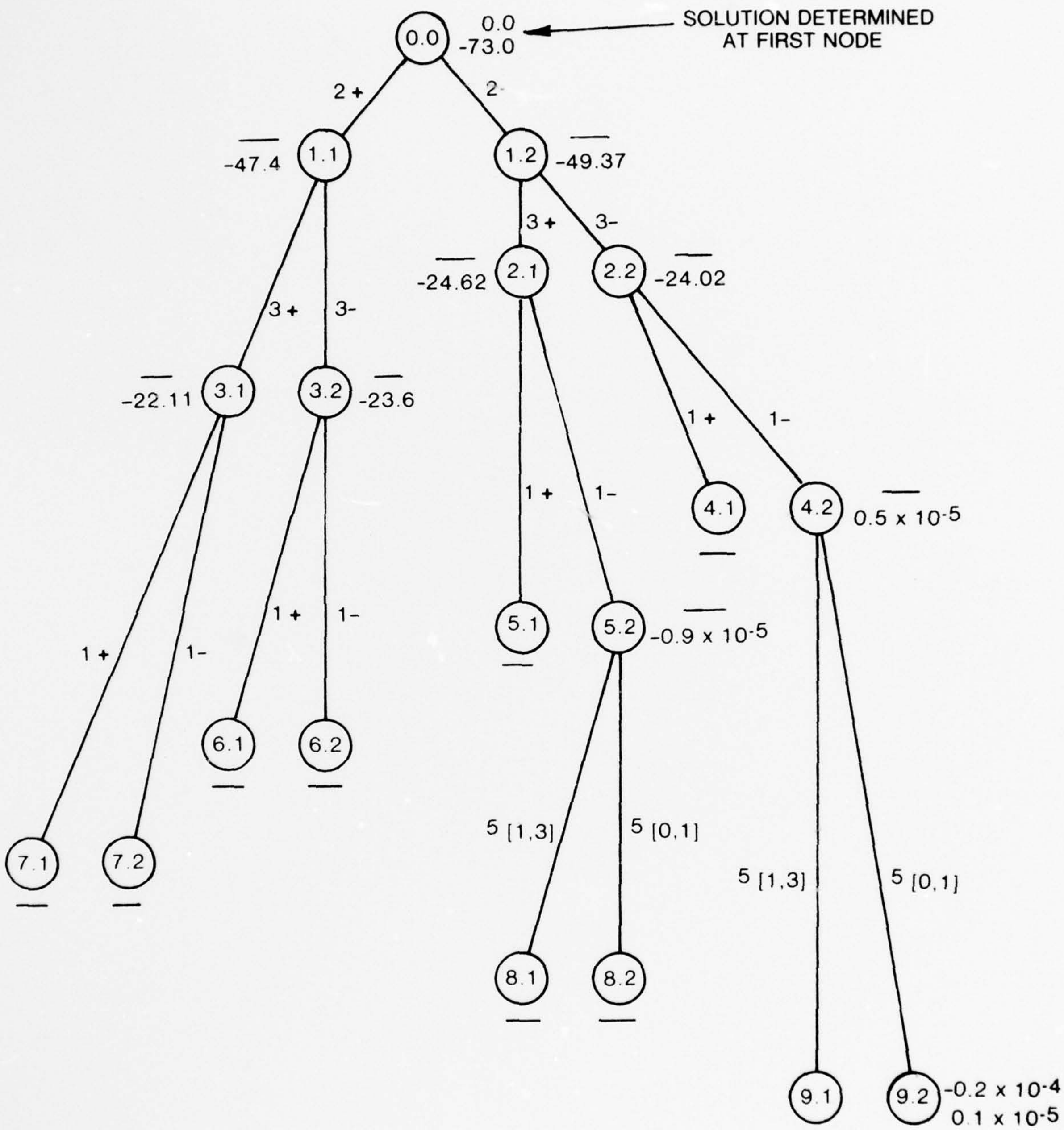

Figure 7. Branch and Bound Tree for Piecewise Linear Market When Forced Past Solution.

necessarily convex) then solving the aggregate of subproblems is tantamount to solving the original problem exactly. Such was the case in both examples 1 and 2. In this example, four of the piecewise linear functions in Figures 5 and 6 have been replaced with smooth counterparts. The new functions were constructed to pass through the points (1,•) and (2,•), and are given by:

$$p_{d_{11}} = 16.516\ e^{(-0.03175x_1)}$$

$$p_{d_{33}} = -0.75\ x_3^2 - 0.25\ x_3 + 8$$

$$p_{s_{11}} = -3.75\ x_1^2 + 14.75\ x_1$$

$$p_{s_{22}} = 17.31234\ \log(x_2^{0.463}+1)$$

Substituting these functions for the originals in problem B leads to a new minimization problem that can be written as

$$\min_{w,x} \sum_{i=1}^{3} (\min(0,w_i) + x_i) \qquad \text{(C)}$$

subject to

$$w_1 + 3.75\ x_1^2 - 13.75x_1 + 16.516\ e^{(-0.03175x_1)} - 4x_2 - 4x_3 = -7$$

$$w_2 - 5x_1 - 17.31234\ \log(x_2^{0.463}+1) - x_2 - 5x_3 = -29$$

$$w_3 - \begin{Bmatrix} 0 & ;\ x_1 \leqq 1 \\ 3x_1-3; & x_1 > 1 \end{Bmatrix} - 3x_2 - \begin{Bmatrix} 0.75x_3^2 + 4.25x_3 - 8 & ;\ x_3 \leqq 1 \\ 0.75x_3^2 - 0.25x_3 - 3.5; & x_3 > 1 \end{Bmatrix} = -4$$

$$3.75\ x_1^2 - 14.75\ x_1 + 16.516\ e^{(-0.03175\ x_1)} - 4x_2 - 4x_3 \leqq -7$$

$$-5x_1 - 17.31234\ \log(x_2^{0.463}+ 1) - 2x_2 - 5x_3 \leqq -29$$

$$\begin{Bmatrix} 0 & ;\ x \leqq 1 \\ 3x_1-3; & x_1 > 1 \end{Bmatrix} - 3x_2 - \begin{Bmatrix} 0.75x_3^2 + 5.25x_3 - 8 & ;\ x \leqq 1 \\ 0.75x_3^2 + 0.75x_3 - 3.5; & x \geqq 1 \end{Bmatrix} \leqq -4$$

$$x_1 \geqq 0\ ,\ x_2 \geqq 0\ ,\ x_3 \geqq 0$$

The number of cuts required to approximate a piecewise linear function exactly is equal to the number of segments constituting that function. When the function is smooth, it cannot be represented exactly by a finite number of linear segments but can be approximated with arbitrary precision by increasing the number of cuts. In this example, six cuts were made in the original problem variables (x_1,x_2,x_3) over the closed interval [0,3]. Owing to the fact that the cuts were evenly spaced every half integer, and that the graphs of the smooth functions pass through the solution points of (B), it is reasonable to expect that the solution to (C) would coincide with one or more of these points. This indeed was the case, the identical solution $x_1^* = (2,1,1)$, resulted for (C). The associated branch and bound tree is shown in Figure 8. The algorithm is seen to have converged in the tenth stage at node 10.1 after 22 subproblems had been solved. This contrasts with the first two examples where the solution occurred on the first iteration; however, in each of these three problems, the first feasible point produced by the algorithm resulted in the solution. Finally, we observe from Figure 8 that at the tenth stage, the best upper bound and the best lower bound are nominally equal implying that the general conditions for optimality have been satisfied, so the algorithm terminated. If an equilibrium point had not yet been found at this stage, it would have been reasonable to conclude that none existed for the given model. The other two equilibrium points were not uncovered.

3.4 Network Traffic Flow - The model of the road system considered here derives from the notion that there exists a large community of users, each of whom takes the quickest route available, given the actions of other users. The number of trips taken is assumed to depend on the time required to make a trip, while the travel time on a particular road is assumed to depend on traffic volume. The example that we will investigate was studied by Asmuth [3] who used stationary point theory in conjunction with the Eaves-Saigal algorithm [9] to obtain a solution. As will be seen, the traffic flow problem closely resembles the multicommodity network presented in example 1.

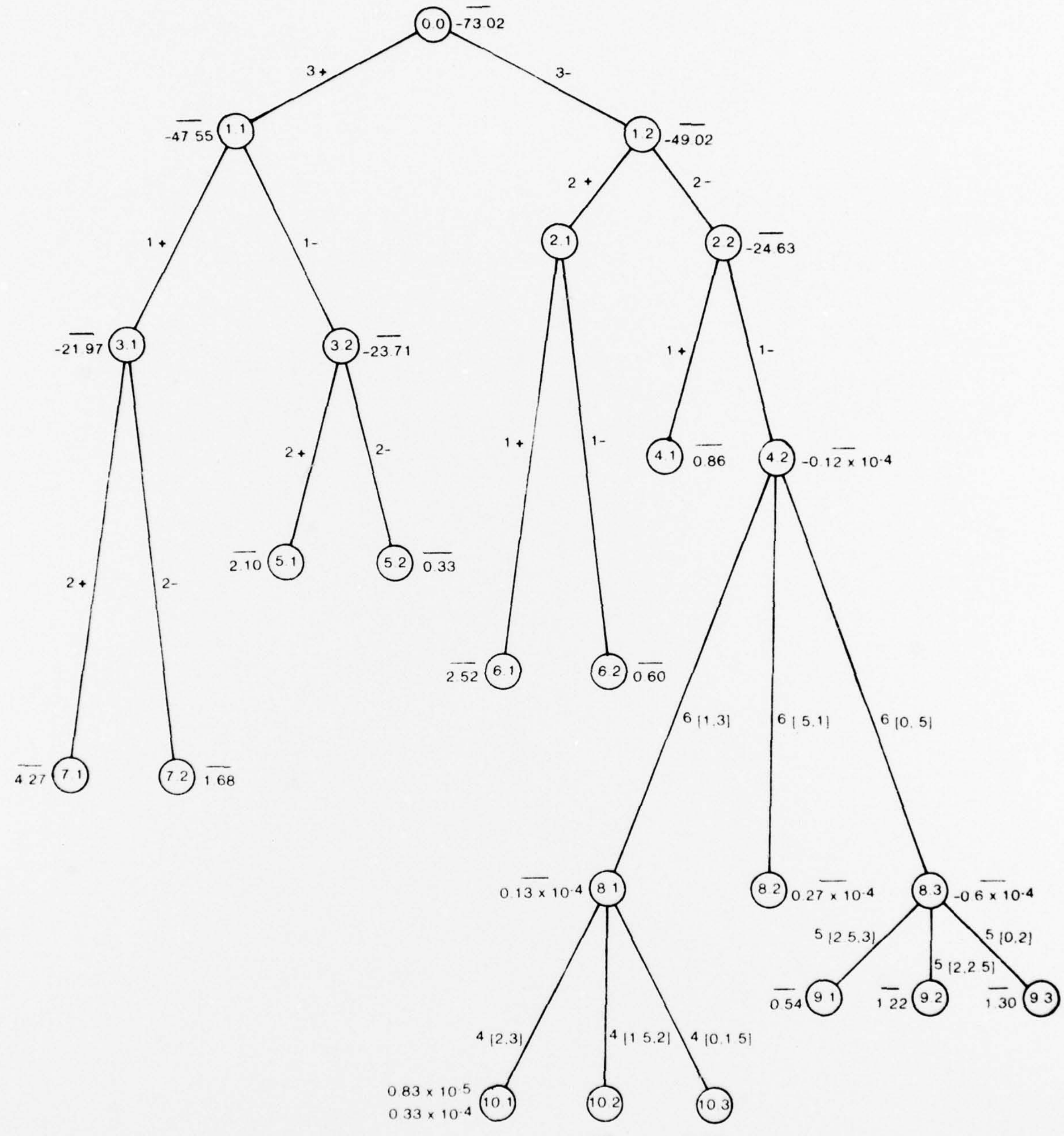

Figure 8. Branch and Bound Tree for General Market

To formulate the model, consider a directed network (N,A) with nodes i in N and arcs ij in A . For each arc ij , we are given a delay function f_{ij} which expresses travel time on arc ij as a function of the traffic flows on the arcs of the network. The travel time along arc ij will necessarily depend on the flow on that arc, but may as well depend on flows along other arcs. For example, a two-way street could be modeled as a pair of opposing arcs where the flow on one of the arcs might affect the travel time on the other.

For each pair of distinct nodes i and k we are also given a travel demand function $g_{i,k}$ which expresses demand for travel from i to k as a function of travel times between nodes on the network. Demand for travel from i to k will depend on travel time from i to k as well as on travel times between other pairs of nodes; for instance, from i to some alternate destination.

Numerous solution procedures have been proposed for computing equilibrium traffic flows and travel times on the network. When f is integrable and g has an inverse which is integrable the usual approach has been to reformulate the equilibrium problem as a convex programming problem. These conditions will be met if each f_{ij} depends only on the total flow along arc ij , and each $g_{i,k}$ is monotone decreasing and depends only on the travel time from node i to node k . Beckman [6], Florian and Nguyen [14], and others have addressed this problem under comparable conditions.

In practice, the demand and delay functions f and g are at best empirical fits and can be endowed with these or any other restrictions which may seem useful. Asmuth's approach does not depend on such restrictions, but only requires that the delay functions f_{ij} be positive on each arc, that the travel demand functions $g_{i,k}$ be nonnegative and bounded for each pair of nodes, and that the network be complete; that is, a directed path must exist from every node to every other node. From a strictly analytic point of view, we will require only that it be possible to put the functions f and g into a separable form. However, if the model is to accurately reflect the properties of the system it might be desirable to adopt the above restrictions.

The mathematical conditions for a user equilibrium are presented below. The travel time from node i to node k will be written as $t_{i,k}$ and the flow on arc ij with destination k will be written as $y_{ij,k}$. It will be said that the travel time vector t and flow vector y are in equilibrium if the following conditions hold:

$$g_{i,k}(t) = \sum_j y_{ij,k} - \sum_j y_{ji,k} \qquad i \neq k,\ i,k \in N \qquad (17)$$

$$y \geqq 0 \qquad (18)$$

$$\begin{aligned} t_{i,k} &\leqq f_{ij}(y) + t_{j,k} && i \neq k,\ ij \in A,\ k \in N \\ t_{k,k} &= 0 && k \in N \end{aligned} \qquad (19)$$

$$y_{ij,k}(f_{ij}(y) + t_{j,k} - t_{i,k}) = 0 \qquad i \neq k,\ ij \in A,\ k \in N \qquad (20)$$

$$y_{ij} = \sum_k y_{ij,k} \qquad ij \in A \qquad (21)$$

Condition (17) is the conservation of flow equation. It says that the traffic leaving node i with destination k is the sum of the traffic arriving at node i with destination k and the traffic originating at i with destination k. Condition (18) says that traffic flows cannot be negative.

Conditions (19) and (20) require that traffic flow by the fastest route available. In (19) we require that $t_{i,k}$ not exceed the minimum travel time from i to k given the flows y on the network; (20) limits the traffic to those routes which achieve this minimum travel time. Together (19) and (20) imply the principle of minimum travel time. This says that if any traffic flows from i to k; that is, if $\sum_j y_{ij,k} > 0$, then

$$t_{i,k} = \min_j (f_{ij}(y) + t_{j,k})$$

Equation (21) relates the basic flows to the total arc flows.

It may be useful to think of this system as a multicommodity network, where all of the traffic destined for a particular node k is a separate commodity, all of which must be shipped to node k via the network. In this way $g_{i,k}(t)$ is the amount of commodity k which must travel from node i to node k. This trip will traverse a path of arcs from i to k.

Conditions (17) - (21) can be put in the form of problem P as follows:

$$\min_{\substack{y\geqq 0\\ t\geqq 0}} \sum_{k\in N} \sum_{\substack{ij\in A\\ i\neq k}} < y_{ij,k} , (-f_{ij}(y) - t_{j,k} + t_{i,k}) >$$

subject to

$$- f_{ij} - t_{j,k} + t_{i,k} \leqq 0 \qquad i\neq k,\ ij\in A,\ k\in N$$

$$g_{i,k} - \sum_j y_{ij,k} + \sum_j y_{ji,k} = 0 \qquad i\neq k,\ i,k\in N$$

$$y_{ij} - \sum_k y_{ij,k} = 0 \qquad ij\in A$$

Rewriting this problem in the form of (1.7) we get

$$\min_{\substack{y\geqq 0\\ t\geqq 0\\ w}} \sum_{k\in N} \sum_{\substack{ij\in A\\ i\neq k}} \{\min(0,w_{ij,k}) + y_{ij,k}\}$$

subject to

$$w_{ij,k} - f_{ij} - t_{j,k} + t_{i,k} + y_{ij,k} = 0 \qquad i\neq k,\ ij\in A,\ k\in N$$

$$- f_{ij} - t_{j,k} + t_{i,k} \leqq 0 \qquad i\neq k,\ ij\in A,\ k\in N$$

$$g_{i,k} - \sum_j y_{ij,k} + \sum_j y_{ji,k} = 0 \qquad i\neq k,\ i,k\in N$$

$$y_{ij} - \sum_k y_{ij,k} = 0 \qquad ij\in A$$

The sample problem presented below is from [4] and is based on the directed network shown in Figure 9.

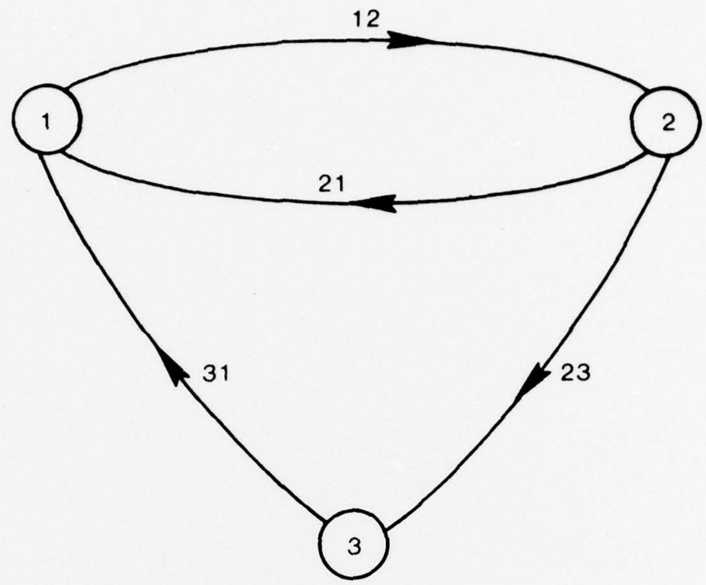

Figure 9. Sample Traffic Network

Here $N = \{1,2,3\}$ and $A = \{12,21,23,31\}$ where arcs 12 and 21 represent a two-way street. The delay functions are

$$f_{12}(y) = 10 + e^{(y_{12}-10)} + 1.25 \log(y_{21}+1.0)$$

$$f_{21}(y) = 10 + e^{(y_{21}-10)} + 1.25 \log(y_{12}+1.0)$$

$$f_{23}(y) = 4 + e^{(y_{23}-12)}$$

$$f_{31}(y) = 4 + e^{(y_{31}-20)}$$

where

$$y_{ij} = \sum_k y_{ij,k}$$

and the travel demand functions are

$$g_{1,2}(t) = \frac{80}{t_{1,2}+1}$$

$$g_{1,3}(t) = \frac{120}{t_{1,3}+1}$$

$$g_{2,1}(t) = \begin{cases} \dfrac{40}{t_{2,1}+1} & \text{if } t_{2,1} \geqq t_{2,3} \\ \dfrac{100}{t_{2,1}+1} & \text{if } t_{2,1} \leqq t_{2,3} \end{cases}$$

$$g_{2,3}(t) = \begin{cases} \dfrac{80}{t_{2,3}+1} & \text{if } t_{2,1} \geqq t_{2,3} \\ \dfrac{20}{t_{2,3}+1} & \text{if } t_{2,1} \leqq t_{2,3} \end{cases}$$

$$g_{3,1}(t) = \frac{60}{t_{3,1}+1}$$

$$g_{3,2}(t) = \frac{100}{t_{3,2}+1}$$

When more than one function value is given at a particular point (e.g., $t_{2,1} = t_{2,3}$) the value of g is the convex hull of the two values. In this case some of the travelers from node 2 will go to either 1 or 3 depending on which is closest. If the travel times are equal then those travelers who want to go to either 1 or 3 will be divided between the two destinations.

In their present form, the demand functions $g_{2,1}$ and $g_{2,3}$ exhibit an implicit dependency on the travel times $t_{2,1}$ and $t_{2,3}$ and therefore, must be made separable before the equilibrium problem can be solved. Although this cannot be done explicitly, the desired result can be achieved by considering the following three disjoint partitions of t :

$$t_{2,1} < t_{2,3} , \quad t_{2,1} = t_{2,3} , \quad t_{2,1} > t_{2,3} .$$

The mathematical program associated with each of these partitions comprises 26 variables and 27 constraints. Of the 26 variables, 12 are of the type required to achieve separability of the functions while the remainder are defined in the original problem statement.

The solution was uncovered in the third partition at the 84th stage after 168 subproblems had been solved, and once again, coincided with the first feasible point found. The resultant branch and bound tree is not displayed because of its extensive length, but the final computations are highlighted in Table 1 along with the results obtained by the Eaves-Saigal algorithm. The minor discrepancies observed between the variable and functional values computed by either method can be attributed to the grid size superimposed on the algorithm and are, hence, subject to control. Finer resolution is strictly a matter of increasing the number of grid points prescribed for the original nonlinear variables and solving a proportionately larger problem.

4.0 Conclusions

The computation of equilibria plays a major role in the analysis of economic and transport systems. Whenever the equilibrium problem can be formulated as a set of complementarity equations, we have shown for those cases where the original functions are implicitly separable, that nonconvex programming can be used to obtain a solution to either problem. A general algorithm based on branch and bound techniques was adapted to perform the equilibrium computation. Unlike the usual nonconvex program though, where the solution is recognized only when equality is achieved between the best upper and best lower bounds, an independent check can be made for the solution at each iteration because the best upper bound is known at the outset. As our computational experience demonstrates, this enhancement markedly increases the efficiency of the algorithm.

Table 1

RESULTS FOR TRAFFIC NETWORK PROBLEM

	Eaves-Saigal Algorithm		MOGG	
i,k	$t_{i,k}$	$g_{i,k}(t)$	$t_{i,k}$	$g_{i,k}(t)$
1,2	19.30	3.94	19.45	3.91
1,3	28.43	4.08	28.58	4.06
2,1	13.22	2.81	13.21	2.81
2,3	9.13	7.90	9.13	7.90
3,1	4.09	11.79	4.09	11.79
3,2	23.38	4.10	23.54	4.07
ij,k	$y_{ij,k}$	f_{ij}	$y_{ij,k}$	f_{ij}
12,2	8.04	19.30	7.99	18.68
12,3	4.08	19.30	4.06	18.68
21,1	1.15	13.22	1.21	13.21
21,3	0.00	13.22	0.00	13.21
23,1	1.66	9.13	1.60	8.76
23,3	11.97	9.13	11.96	8.76
31,1	13.46	4.09	13.41	4.08
31,2	4.10	4.09	4.08	4.08

However, the fact that a numerical procedure will terminate with the correct answer in a finite number of iterations is no guarantee that it will be of any practical use. The combination of method and algorithm under study derives its tentative usefulness from the observation that for most problems investigated, convergence occurred in a far smaller number of iterations than theoretically possible. The results have been especially encouraging for problems of larger dimensions; and in all cases, the equilibrium solution coincided with the first feasible point found by MOGG.

The affine equilibrium problem or linear complementarity problem holds a particular interest because of its unique structure and implicit relationship to an equivalent linear program. Because of the similarity between the first subproblem set up by MOGG and this linear program, immediate solutions are often obtainable from MOGG at little extra cost. In fact, the additional work required to determine the equivalent linear program, even for relatively small problems, is often more burdensome and more computationally expensive than permitting MOGG to run beyond its first subproblem to a point of convergence. A further and decided advantage of MOGG is that it will solve all affine equilibrium problems regardless of their matrix structure. By contrast, the majority of alternative procedures available are limited in their application to a number of special cases which do not necessarily arise in practice.

REFERENCES

[1] ARROW, K. J. and G. DEBREU (1954). Existence of an equilibrium for a competitive economy. Econometrica 22 265-90.

[2] ARROW, K. and F. HAHN (1971). General Competitive Analysis. Holden-Day Publishing Company, San Francisco.

[3] ASMUTH, R. L. (1978). Traffic network equilibria. Technical Report SOL 78-2. Department of Operations Research, Stanford University.

[4] ASMUTH, R. L., B. C. EAVES and E. L. PETERSON (1978). Studying economic equilibria on affine networks via Lemke's algorithm. Discussion Paper No. 314. Department of Operations Research, Stanford University.

[5] BEALE, E. M. L. and J. A. TOMLIN (1970). Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables. Proceedings of the Fifth International Conference on Operations Research (J. Lawrence, ed.) 447-454. Tavistock Publications, London.

[6] BECKMAN, M. J. (1967). On the theory of traffic flow in networks. Traffic Quarterly 21 109-116.

[7] BRACKEN, J. and J. E. FALK (1976). Computation of particular economic equilibria. IDA Paper P-1208. Institute for Defense Analysis, Arlington, Virginia.

[8] EAVES, B. C. (1971). The linear complementarity problem. Management Sci. 17 612-34.

[9] EAVES, B. C. and R. SAIGAL (1972). Homotopies for computation of fixed points on unbounded regions. Mathematical Programming 3 (2) 225-237.

[10] ELKEN, T. (1977). The computation of economic equilibria by path methods. SOL Technical Report 77-26. Department of Operations Research, Stanford University.

[11] FALK, J. E. (1969). Lagrange multipliers and nonconvex programs. SIAM J. Control 7 534-545.

[12] FALK, J. E. (1972). An algorithm for locating approximate global solutions of nonconvex, separable problems. Technical Paper Serial T-262. Program in Logistics, The George Washington University.

[13] FALK, J. E. and R. M. SOLAND (1969). An algorithm for separable nonconvex programming problems. Management Sci. 15 550-569.

[14] FLORIAN, M. and S. NGUYEN (1974). A method for computing network equilibrium with elastic demands. Transportation Sci. 8 (4) 321-332.

[15] GROTTE, J. H. (1976). Program MOGG - A code for solving separable nonconvex optimization problems. IDA Paper 1318. The Institute for Defense Analysis, Arlington, Virginia.

[16] KUHN, H. W. (1956). On a theorem of Wald. Linear Inequalities and Related Systems (H. W. Kuhn and A. W. Tucker, eds.) 265-273. Princeton University Press.

[17] LEMKE, C. E. (1965). Bimatrix equilibrium points and mathematical programming. Management Sci. **11** 681-89.

[18] MANGASARIAN, O. L. (1976). Characteristization of linear complementarity problems as linear programs. Computer Sciences Technical Report #271. Computer Science Department, University of Wisconsin - Madison.

[19] MILLER, C. E. (1963). The simplex method for local separable programming. Recent Advances in Mathematical Programming (R. L. Graves and P. Wolfe, eds.) 89-100 McGraw Hill, New York.

[20] NASH, J. F. (1950). Equilibrium points in n-person games. Proceedings of the National Academy of Sciences of the U.S.A. **36** 48-49.

[21] NEGISHI, T. (1962). Monopolistic competition and general equilibrium. Review of Economic Studies **28** (3) 196-201.

[22] QUIRK, J. R. and R. SAPOSNIK (1968). Introduction to Equilibrium Theory and Welfare Economics. McGraw-Hill, New York.

[23] SCARF, H. (1967). The approximation of fixed points of a continuous mapping. SIAM J. Applied Math. **15** 1328-43.

[24] SCARF, H. (1973). The Computation of Economic Equilibria. Yale University Press, New Haven.

[25] SOLAND, R. M. (1971). An algorithm for separable nonconvex programming problems II: Nonconvex constraints. Management Sci. **17** 759-773.

[26] TAKAYAMA, T. and G. G. JUDGE (1971). Spatial and Temporal Price and Allocation Models. North-Holland Publishing Company, Amsterdam.

[27] WALRAS, L. (1874). Elements d'Economie Politique Pure, Lausanne: Corbaz. (Translated as Elements of Pure Economics, W. Jaffe, translator, London: Allen and Unwin, 1954.)

[28] WILSON, R. (1978). The bilinear complementarity problem and competitive equilibria of linear economic models. Econometrica **46** (1) January.